



UNIVERSITY OF
ALBERTA

CMPUT 497/500

Foundations of Program Analysis

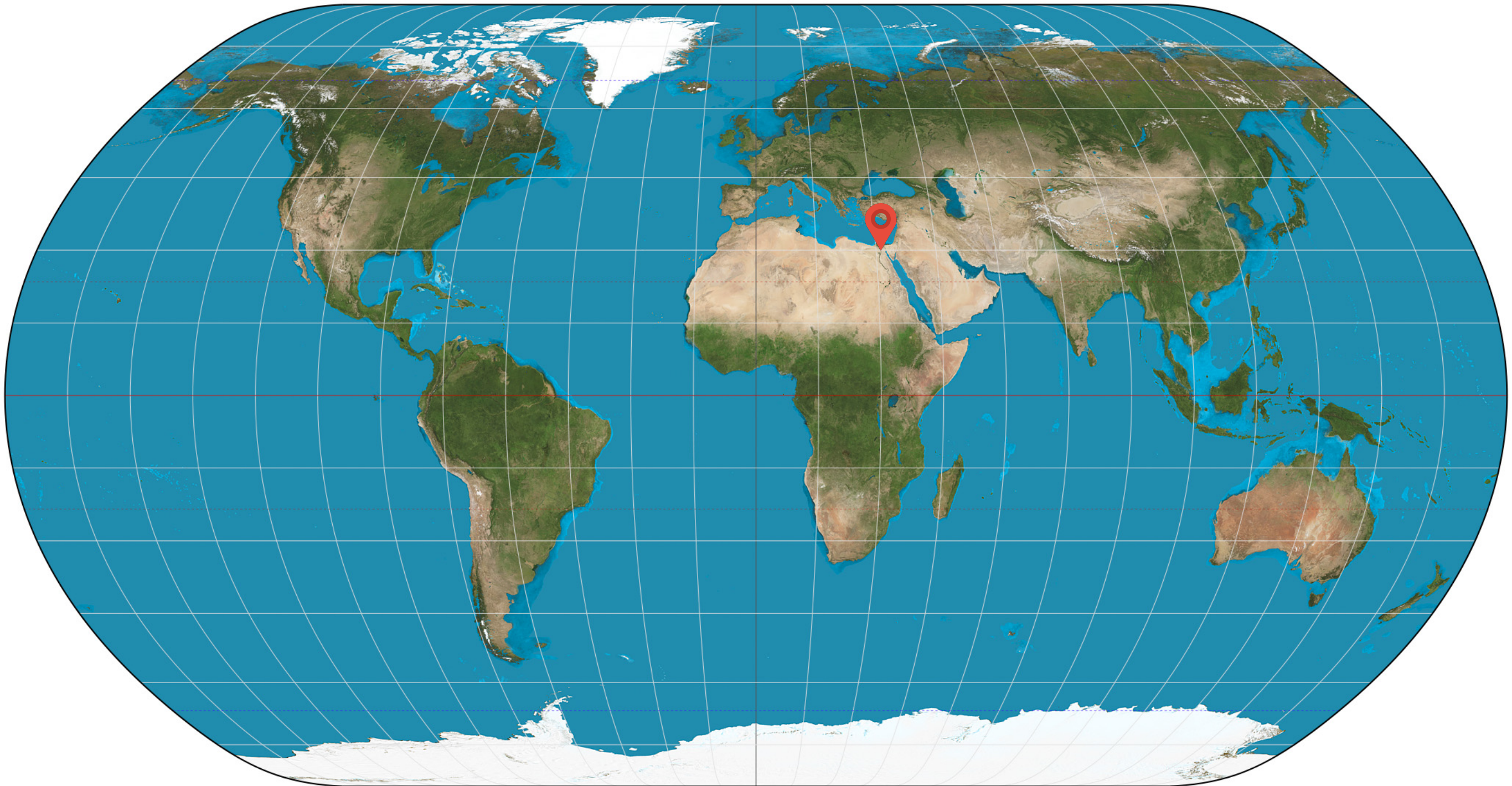
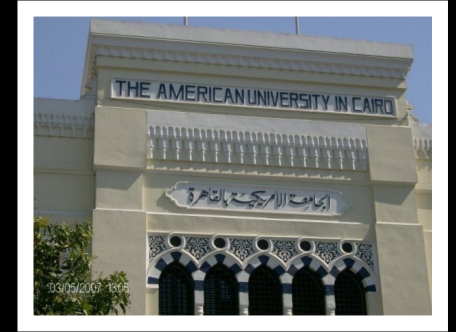
Karim Ali

[@karimhamdanali](https://twitter.com/karimhamdanali)

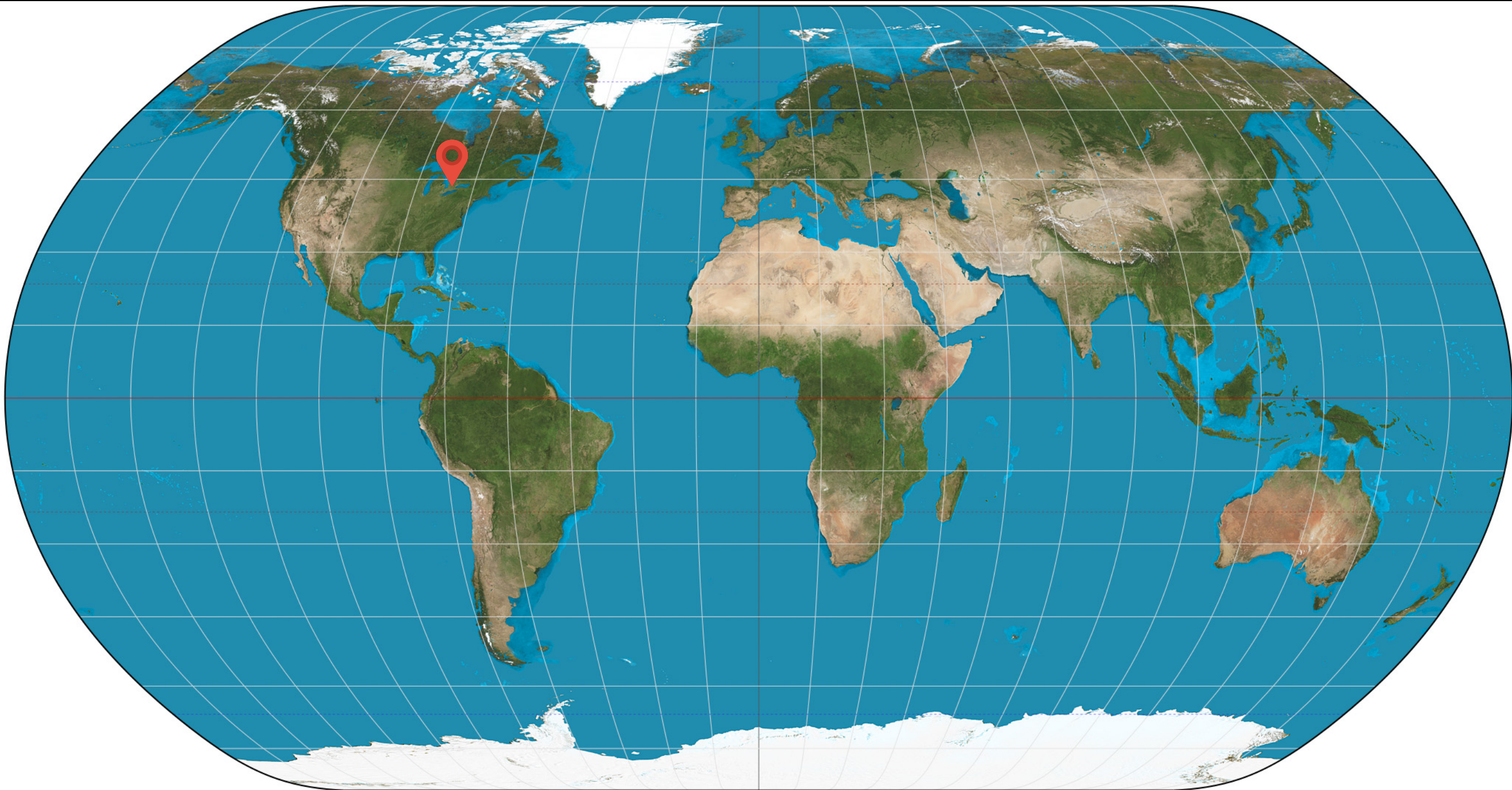
September 3, 2019

GSB 5-3

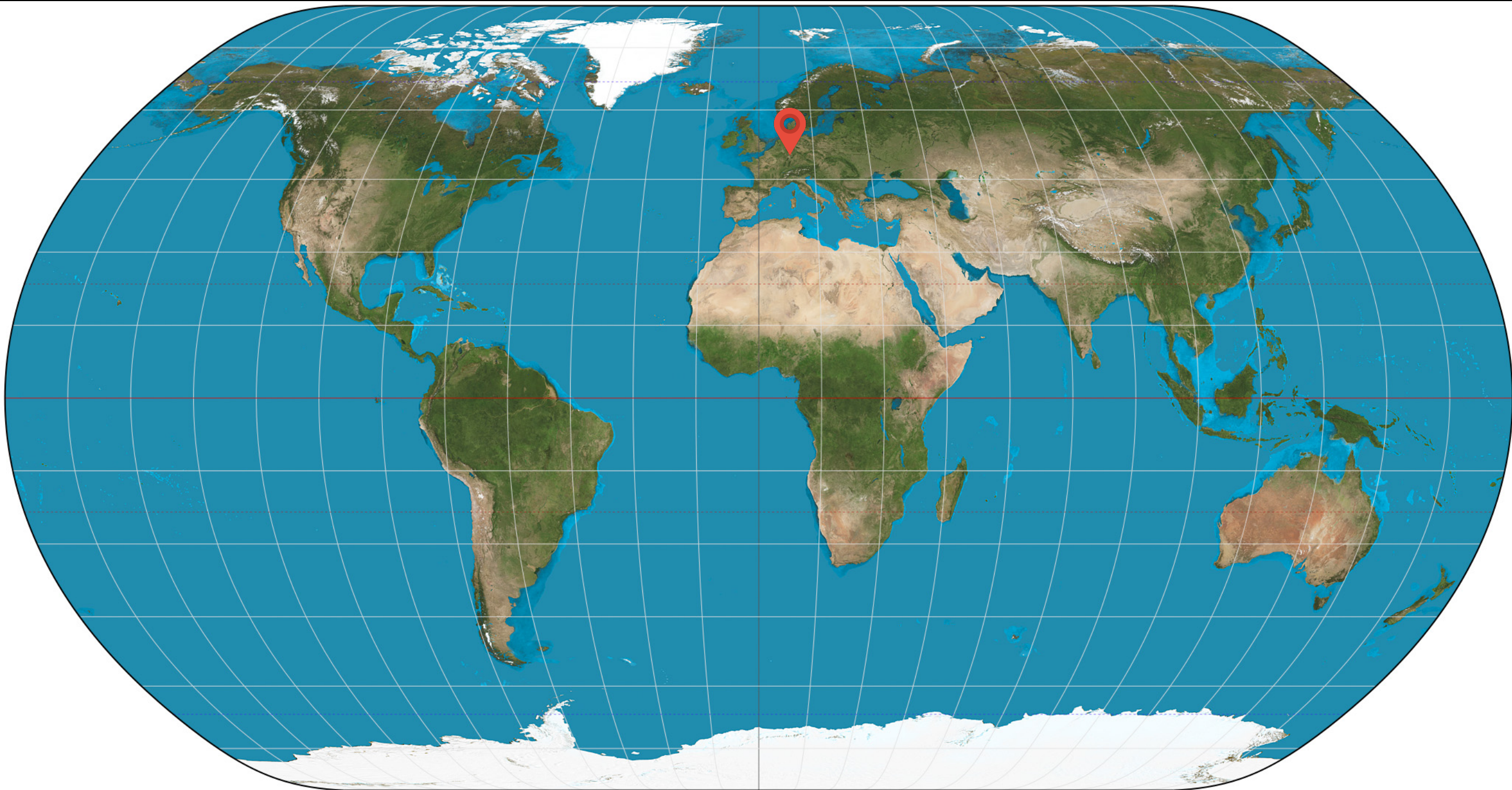
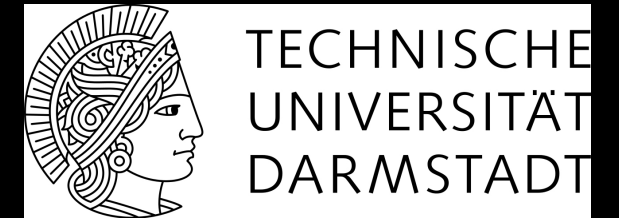
Bachelor's in Egypt 2003—2007



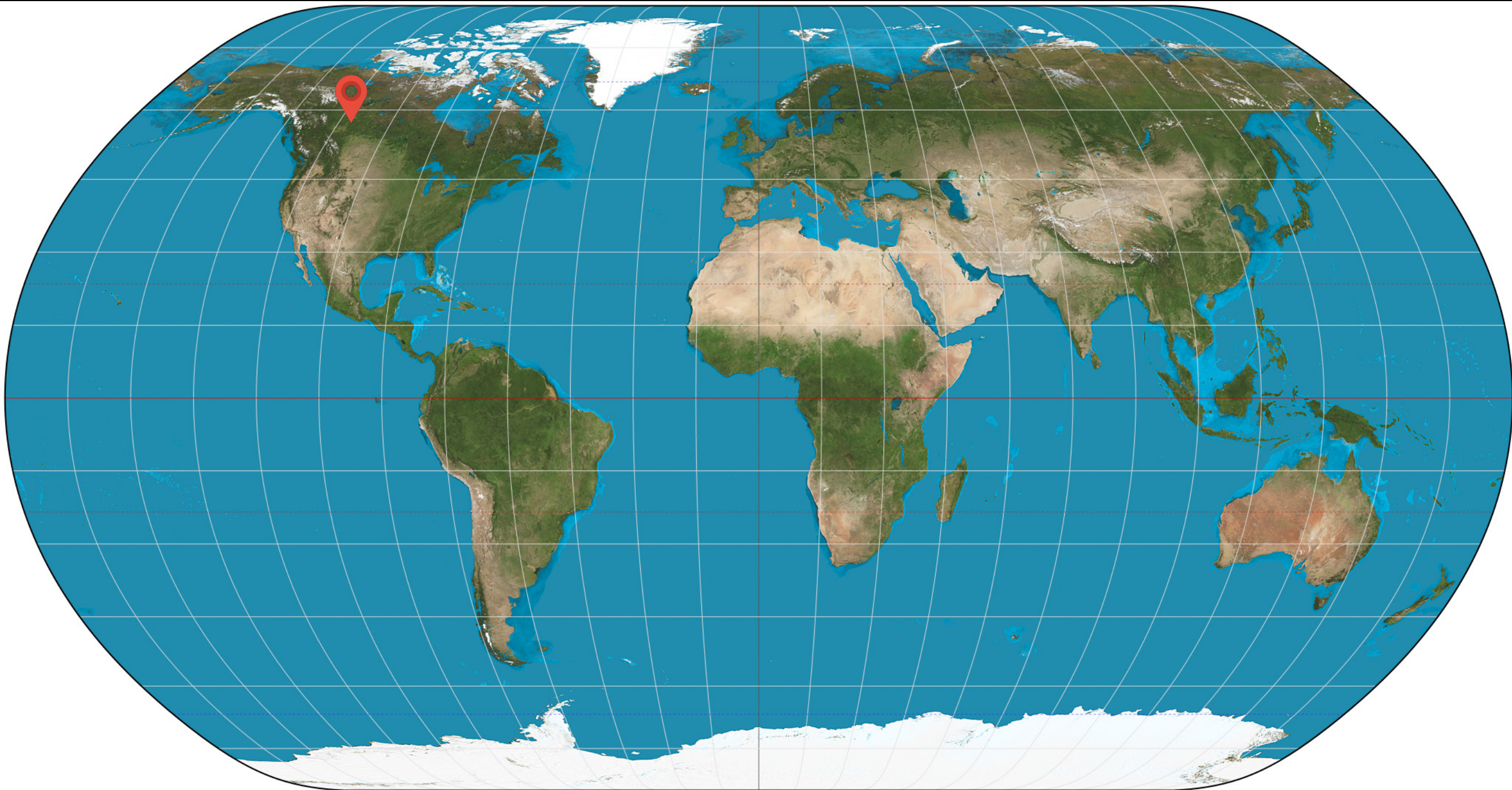
Master's & Ph.D. in Canada 2008—2014



Postdoc in Germany 2014—2016



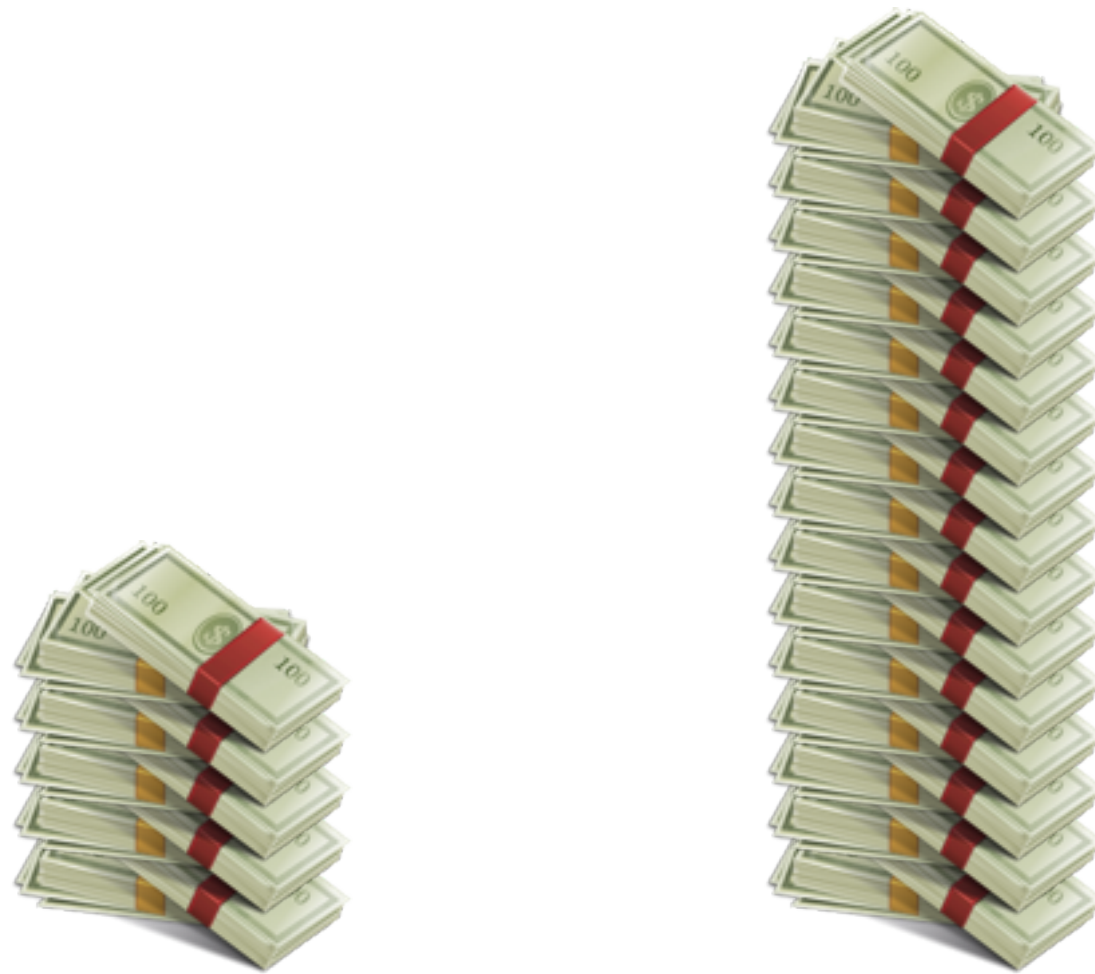
Assistant Professor
2017—???



What about you?

Cost of Software Bugs?





```

static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                uint8_t *signature, UInt16 signatureLen)
{
    OSStatus      err;

    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    err = sslRawVerify(ctx,
                      ctx->peerPubKey,
                      dataToSign,          /* plaintext */
                      dataToSignLen,       /* plaintext length */
                      signature,
                      signatureLen);

    if(err) {
        sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                   "returned %d\n", (int)err);
        goto fail;
    }

fail:
    SLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}

```



```

static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                uint8_t *signature, UInt16 signatureLen)
{
    OSStatus      err;

    ...

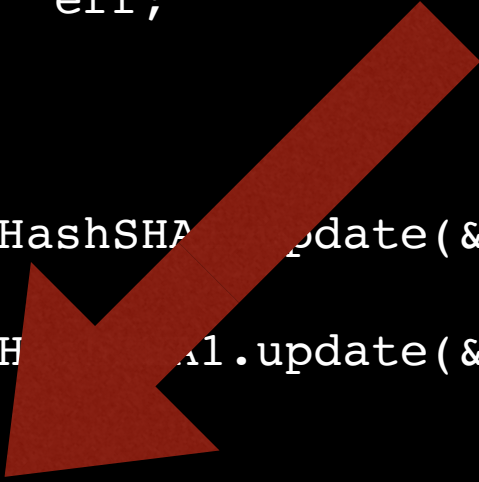
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    err = sslRawVerify(ctx,
                      ctx->peerPubKey,
                      dataToSign,           /* plaintext */
                      dataToSignLen,        /* plaintext length */
                      signature,
                      signatureLen);

    if(err) {
        sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                   "returned %d\n", (int)err);
        goto fail;
    }

fail:
    SLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}

```



```

static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                uint8_t *signature, UInt16 signatureLen)
{
    OSStatus      err;

    ...

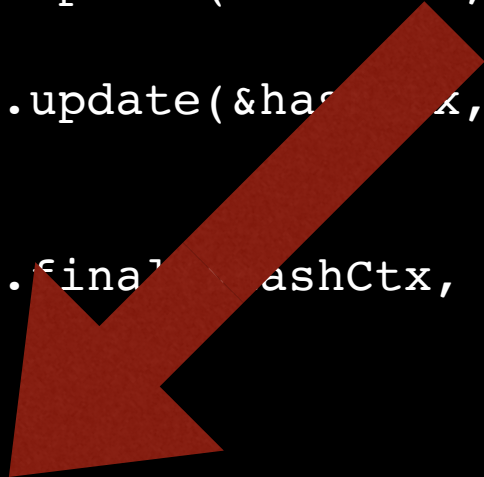
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    err = sslRawVerify(ctx,
                      ctx->peerPubKey,
                      dataToSign,           /* plaintext */
                      dataToSignLen,        /* plaintext length */
                      signature,
                      signatureLen);

    if(err) {
        sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                   "returned %d\n", (int)err);
        goto fail;
    }

fail:
    SLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}

```




```

static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                uint8_t *signature, UInt16 signatureLen)
{
    OSStatus      err;

    ...


    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    err = sslRawVerify(ctx,
                      ctx->peerPubKey,
                      dataToSign,           /* plaintext */
                      dataToSignLen,       /* plaintext length */
                      signature,
                      signatureLen);

    if(err) {
        sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                   "return %d\n", (int)err);
        goto fail;
    }

fail:
    SLFreeBuffer(signedHashes);
    SSLFreeBuffer(hashCtx);
    return err;
}

```



Apple's goto fail!

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                uint8_t *signature, UInt16 signatureLen)
{
    OSStatus      err;

    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    err = sslRawVerify(ctx,
                      ctx->peerPubKey,
                      dataToSign,           /* plaintext */
                      dataToSignLen,        /* plaintext length */
                      signature,
                      signatureLen);

    if(err) {
        sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                   "returned %d\n", (int)err);
        goto fail;
    }

fail:
    SLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```


Apple's goto fail!

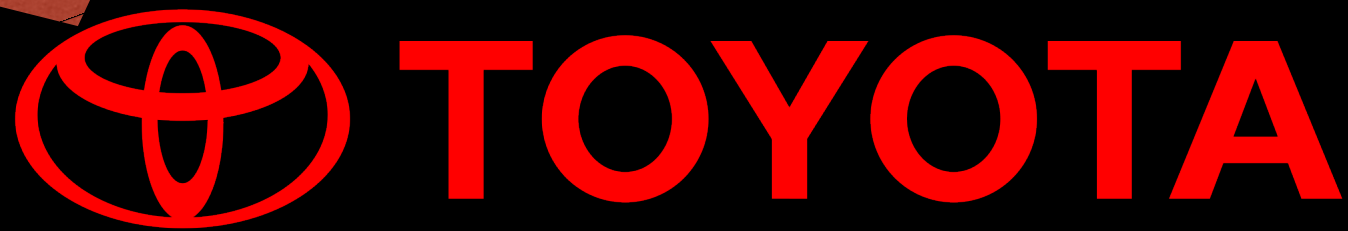
```
static OSStatus  
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,  
                                uint8 *signature, UInt16 signatureLen)
```



```
return err;
```

```
}
```

US \$3 Billion





Medtronic

Alleviating Pain • Restoring Health • Extending Life



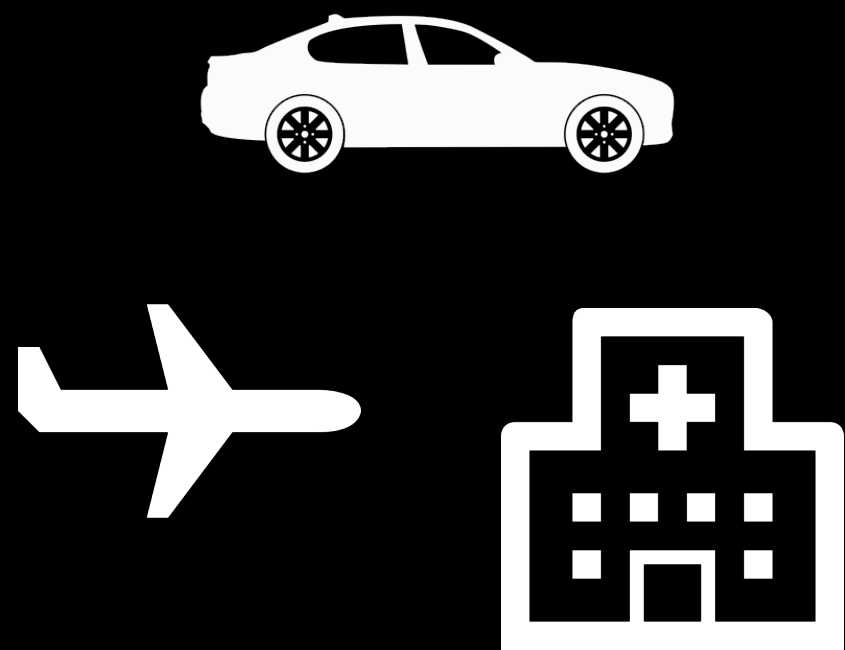
Program Analysis



What is Program Analysis?

Program Analysis

A way of understanding the runtime behaviour of a program without necessarily executing it



Code Navigation

Code Refactoring

Code Recommenders

Program Analysis

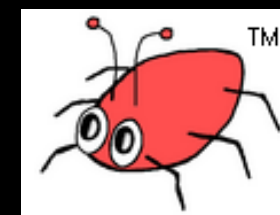
Parallelization

Constant Propagation

Static Inlining

Dead Code Elimination

 CHECKMARX



 **coverity**[®]
A Synopsys Company

 **FORTIFY**

 **AbsInt**

Topics

- Intro + Fundamentals
 - IR
 - Intra-Procedural Analysis
 - Call Graphs
 - Pointer Analysis
 - IFDS/IDE
 - Usability of Analysis Tools
- Seminars
 - Developer Support
 - Android Security
 - Distributed Static Analysis
 - Dynamic Languages
 - Probabilistic Programs
 - Program Synthesis

This Course

- Mix of lectures and in-class hands-on
- Assignments
- Seminars
- Course project

eClass

<http://bit.ly/cmput497-fa19>